

COMP 690 Data Fusion Fall 2008 Exam 1—Solutions

All problems are equally weighted.

1. Write a Python script that reads a line at a time from the file `text.txt` and builds up a dictionary giving the number of times the various words occur in the file. To make things easy, all characters are either lowercase letters or whitespace. You do not have to write the code that outputs the contents of the dictionary.

Hint: Consider the string method `split()` and the dictionary method `has_key()`

As an example, suppose the following is the content of `text.txt`

```
the cat sat on the mat
a dog chased the cat
the cat chased a bird
```

Then the dictionary will contain the following associations.

```
'a':      2
'on':     1
'mat':    1
'chased': 2
'dog ':   1
'cat':    3
'the':    4
'bird':   1
'sat':    1
```

Answer

```
f = open('text.txt')
dict = {}

for line in f:
    words = line.split()
    for word in words:
        if dict.has_key(word):
            dict[word] += 1
        else:
            dict[word] = 1
```

2. Write a function `sum_embedded()` that takes a list of strings. Each string contains at most one numeric substring consisting of a sequence of one or more digits. Examples include 'ab12c' (contains the substring '12'), 'dfe' (contains no such substring), 'mn8' (contains '8'), and '3xy' (contains '3'). The function extracts all the numeric substrings, converts them to integers, and returns their sum. For example,

```
sum_embedded(['ab12c', 'dfe', 'mn8', '3xy'])
```

returns 23.

Answer

```
def sum_embedded(words):
    p = re.compile(r'^\D*(\d+)\D*$')
    sum = 0

    for word in words:
        m = p.match(word)

        if m:
            sum += int(m.group(1))

    return sum
```

3. Define a class `Adder` that contains instance property `sum`, initialized to 1. If an instance is printed, as in

```
>>> print adder1
```

only the value of `sum` is output. Define an instance method `getnum()` that takes no arguments. It prompts for and reads in a string representation of an integer, which it converts to an integer and adds to `sum`. It catches two exceptions: `IOError` and `ValueError`. In both cases, it outputs a message saying which error occurs (and does not update `sum`).

Answer

```
class adder:
    def __str__(self):
        return str(self.sum)
    def __init__(self):
        self.sum = 1
    def get_num(self):
        try:
            self.sum += int(raw_input("Please enter a number: "))
        except IOError:
            print "I/O error"
        except ValueError:
            print "Value error"
```